

Trabalho 3 - Caminho mais curto

Disciplina: Projeto e Análise de Algoritmos

Aluno: Diego S. Seabra

Matrícula: 0040251

Como rodar

Requisitos: **Python v3.9.1**

Para rodar o projeto, execute o comando abaixo passando *via linha de comando*, como *parâmetro*, o nome do arquivo desejado.

```
python3 trabalho3.py transacoesBitcoin.txt
```

ou

```
python trabalho3.py transacoesBitcoin.txt
```

Outras informações serão solicitadas em **runtime**.

Arquivos Disponíveis

Abaixo uma lista dos arquivos disponíveis neste repositório que podem ser utilizados para testes.

Não ponderados (sem peso)

- `naoPonderado.txt` (simples, com poucos dados)
- `estradaCalifornia.txt` (completo, com muitos dados)

Ponderados (com peso)

- `estadosBrasil.txt` (simples, com poucos dados)
- `transacoesBitcoin.txt` (completo, com muitos dados)

Algumas observações/explicações

- O código reconhece automaticamente o tipo de grafo (ponderado ou não ponderado), então esta informação não precisa ser explicitada. É necessário informar **apenas** o nome do arquivo
- Os dados dos arquivos devem estar espaçados em **TABS** ao invés de espaços (os arquivos deste repositório já se encontram com esta configuração)
- Foi utilizado o algoritmo de Dijkstra para o cálculo do caminho mínimo e como este algoritmo é usado normalmente com pesos, para os casos não ponderados assumimos que o peso é 0 e executamos o algoritmo normalmente
- Há dois cálculos de tempo de execução
- Um para cada operação de cálculo do caminho mínimo
- Um para a operação por completo
- Ao rodar o algoritmo, o usuário tem duas escolhas
- Digitar o vértice de *origem* **E** o vértice de *destino*
 - Neste caso o algoritmo vai calcular o caminho mínimo entre dois vértices A e B (origem e destino, respectivamente)
- Ou digitar apenas o vértice de *origem*
 - Neste caso o algoritmo vai calcular o caminho mínimo entre o vértice informado (origem) e **todos** os outros vértices do grafo
- Para mais informações, consulte os comentários no próprio código fonte